REMARKS

This Amendment is filed in response to the Office Action mailed July 8, 2009. All objections and rejections are respectfully traversed.

Claims 1-5 and 30-67 are in the case.

No new claims have been added.

Claims 1-2, 30, 32-35, 38, and 40-44 have been amended.

Interview Summary

Applicant would like to thank Examiner Colan for conducting the Applicant Initiated Interview on August 27, 2009 and for helping to advance this Application closer to allowance. Generally, as will be elaborated upon in greater detail below, the issue discussed involved Applicant's use of a Network File System (NFS) operation. Specifically, Applicant discussed how neither prior art reference (i.e., Chandrashekhar and Ryuutou) discloses an ***NFS*** file handle. More to the point, Applicant discussed that neither prior art reference disclosed ***inserting encryption key metadata into an NFS file handle*** and then **sending the NFS file handle *with the encryption key metadata inserted in the NFS file handle* to the client as a reply to the first NFS operation**. Examiner noted that should Applicant's assertions be correct, the claims as discussed would overcome the references. However, Examiner also noted that a closer look at both references would be performed to verify Applicant's assertions and that a new search would be conducted.

Rejections Under 35 U.S.C. §103

At Paragraph 6 of the Office Action, claims 1-5 and 30-67 were rejected under 35 U.S.C. §103(a) as being anticipated by Chandrashekhar et al., U. S. Patent Publication 2005/0033988 published on February 10, 2005 (hereinafter "Chandrashekhar"), and in view of Ryuutou et al., U.S. Patent Application Publication No. 2002/0083191 published on June 27, 2002 (hereinafter "Ryuutou").

Applicant's claimed novel invention, as set out in representative claim 1, comprises in part:

> 1. A method for establishing identity in a file system, comprising:
>
> receiving, from a client, a first Network File System (NFS) operation concerning an indicated file, the first NFS operation received by a proxy;
>
> forwarding the first NFS operation from the proxy to be received by a file server;
>
> returning a NFS file handle associated with the first NFS operation from the file server to the proxy in response to the file server receiving the first NFS operation from the proxy;
>
> **inserting, by the proxy, metadata into the NFS file handle** in response to receiving the NFS file handle from the file server, **wherein *the metadata is an encryption key***;
>
> **sending**, by the proxy in response to receiving the NFS file handle from the file server, **the NFS file handle *with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation***; and
>
> using, by the client, the metadata and the NFS file handle in a second NFS operation to identify the client and the indicated file.

Chandrashekhar discusses processing file requests sent by a client and received by a proxy using security applications to encrypt, decompress, verify, and decrypt network data by a server receiving the files from the proxy [0058; 0071]. Header policy information is determined, generated, and then stored on the filer server [0055; Fig. 4-5]. However, any metadata added to a file is stripped off before the file data/file attributes are returned to the client [0038].

Ryuutou discloses, in relevant part as cited by Examiner, a client establishing an HTTP connection between the client and a proxy server by initiating a communication connection request [0072-0073]. A session ID is added to header information of an HTTP request to determine whether or not a connection corresponding to a particular series of communications have been established [Abstract; 0017; see also Fig. 10 for an example of an HTTP header information format]. This information about the client is stored in a memory table on the proxy server [0057].

Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Ryuutou, does not disclose Applicant's claimed novel and non-obvious use of

inserting, by the proxy, metadata into the NFS file handle, wherein ***the metadata is an encryption key***; and

sending the NFS file handle ***with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation.***

Applicant claims, in part, a proxy receiving from a client ***a first Network File System (NFS) operation*** concerning an indicated file and forwarding the first ***NFS operation*** from the proxy to be received by a file server. Applicant further claims returning a NFS file handle associated with the first ***NFS operation*** from the file server to the proxy in response to the file server receiving the first NFS operation from the proxy. Applicant further claims **inserting, by the proxy, metadata into the NFS file handle, wherein** ***the metadata is an encryption key***. With that being said, after inserting (the encryption key) metadata into the ***NFS file handle***, Applicant further claims **sending the NFS file handle** ***with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation.***

Applicant respectfully argues that Chandrashekhar does not teach or suggest Applicant's claimed novel **sending a NFS file handle** ***with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation.*** It should first be noted that while Chandrashekhar does disclose adding header information on a file by file basis, there is no indication that Chandrashekhar shows or suggests adding header information to a file handle. However, even if it is assumed *arguendo* that Chandrashekhar does disclose a file handle, Chandrashekhar strips off any metadata added to a file before the file data/file attributes are returned to the client. In contrast, Applicant claims sending a NFS file handle ***with the metadata inserted in the NFS file handle to the client*** as a reply to the first NFS operation. As such, because Chandrashekhar strips off any metadata (i.e., any potential encryption key data) even if it

is assumed *arguendo* that Chandrashekhar shows a file handle, Chandrashekhar must still be silent to Applicant's claimed novel **sending a NFS file handle *with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation***.

Applicant respectfully argues that Ryuutou does not teach or suggest Applicant's claimed novel **sending a NFS file handle with the metadata inserted in the NFS file handle to the client...wherein *the metadata is an encryption key***. It should first be noted that while Ryuutou shows adding a session ID to header information which is sent back to the client, similarly to Chandrashekhar, Ryuutou is silent to adding the session ID to a file handle or an NFS file handle. However, even if it is assumed *arguendo* that Ryuutou shows adding a session ID to a file handle or NFS file handle, Ryuutou is still be silent to Applicant's claimed sending a NFS file handle with the metadata inserted in the NFS file handle to the client...**wherein *the metadata is an encryption key***. Specifically, Ryuutou shows adding a session ID as header information which is sent back to the client. A session ID is not an encryption key. In contrast, Applicant claims inserting metadata into the NFS file handle, **wherein *the metadata is an encryption key***. As such, because a session ID is not the same as an encryption key, Ryuutou is also silent to Applicant's claimed novel **sending a NFS file handle with the metadata inserted in the NFS file handle to the client...wherein *the metadata is an encryption key***.

Additionally, Applicant respectfully argues that Ryuutou does not teach or suggest Applicant's claimed novel **sending a NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation***. Specifically, while Ryuutou shows a client establishing an HTTP protocol communication connection request with a proxy server, Ryuutou shows adding a session ID to header information in response to the HTTP protocol communication connection request which is not an NFS operation. In contrast, Applicant claims sending a NFS file handle with the metadata inserted in the NFS file handle to the client ***as a reply to the first NFS operation***. As such, because Ryuutou is silent to the concept of NFS or an NFS operation (since Ryuutou is discussed in terms of responding to an HTTP protocol

communication connection request which is not an ***NFS operation***), Ryuutou must be silent to Applicant's claimed **sending a NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation***.

Accordingly, Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Ryuutou, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103. Chandrashekhar and Ryuutou, taken singly or in any combination, does not disclose Applicant's claimed novel and non-obvious use of

inserting, by the proxy, metadata into the NFS file handle, wherein *the metadata is an encryption key*; and

sending the NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation*.

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art references was derived, in part, from the following excerpts:

Chandrashekhar

> [0038]…The meta-data relates to key management, length of the original file/dataset, whether the file was compressed prior to encryption or not, integrity checks for file data. The meta-data is stripped off before the file data/file attributes are returned to the client… (emphasis added)

Ryuutou

> [0017] A communication distribution controlling method according to a first preferred embodiment of the present invention is a communication distribution controlling method distributing one communication to any of a plurality of relay devices, which can relay the one communication, in correspondence with a connection request of the one communication within a series of communications from a client. With this method, a communication connection request is received from a client, whether or not a communication connection corresponding to a series of communications is established is determined according to an identifier written in the communication connection request, and the requested communication is connected to a particular relay device as a relay destination of an established communication connection, if the communication connection is established. (emphasis added)

[0057] FIG. 6 is a flowchart showing the process of a communication connection management method in this preferred embodiment. In FIG. 5, when a new communication connection to a gateway is established in correspondence with the initial communication connection request within one session, and a session ID is set, its contents are stored in a memory (table) not shown. At the same time, a timer not shown is started, and its elapsed time is monitored. (emphasis added)

[0072] As explained with reference to FIG. 9, the session number S4, and the session ID ZZZ are set by the proxy in correspondence with this communication connection request. The newly set session ID is added to the header information, for example, within the reply (1) to the PC-A 31a in FIG. 4, and returned from the proxy 32a to the client side. (emphasis added)

[0073] At this time, ZZZ as the session ID is added between B and C in the header information shown in FIG. 10. As a method adding a session ID, a method such as Netscape Cookie, with which a browser side can recognize and store, for example, data that is additionally described in an HTTP header, is used. (emphasis added)

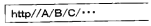

FIG. 10

[0074] A reply including header information to which a session ID is added is returned from a proxy side to a PC side as described above, so that header information including the session ID can be used as the header information in the second and subsequent communication connection requests. (emphasis added)

Conclusion

All new claims and/or claim amendments are believed to be fully supported by Applicant's specification.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Michael T. Abramson/
Michael T. Abramson
Reg. No. 60,320
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927